SCRAM Evaluation Key Principles

- Minimal Disruption
  - Artefact Review (plans, procedures, model evidence) conducted offline
  - Information is collected one person at a time
  - Interviews typically last an hour
- Independent
  - Evaluation team members are organisationally independent of the program under review
    - Some SCRAM reviews have been joint contractor/customer team – facilitates joint commitment to resolve review outcomes
- Non-advocate
  - All significant issues and concerns are considered and reported regardless of origin or source (Customer and/or Contractor).

SCRAM Evaluation Key Principles

- Non-attribution
  - Information obtained is not attributed to any individual
  - Focus is on identifying and mitigating the issues/risk
- Corroboration of Evidence
  - Significant Findings and Observations based on at least two independent sources of corroboration
- Rapid turn-around
  - One to two weeks spent on-site
  - Executive out-briefing presented at end of second week
  - Written report two weeks later

SCRAM Evaluation Key Principles

- Sharing Results, Openness and Transparency
  - For the parametric modelling component of a SCRAM assessment, organisation under review may witness data analysis and challenge results
  - Preliminary out brief of findings is delivered prior to departure from evaluation site
  - Builds cooperation and trust
  - Builds confidence in the schedule forecast
  - SCRAM Team is the final arbiter

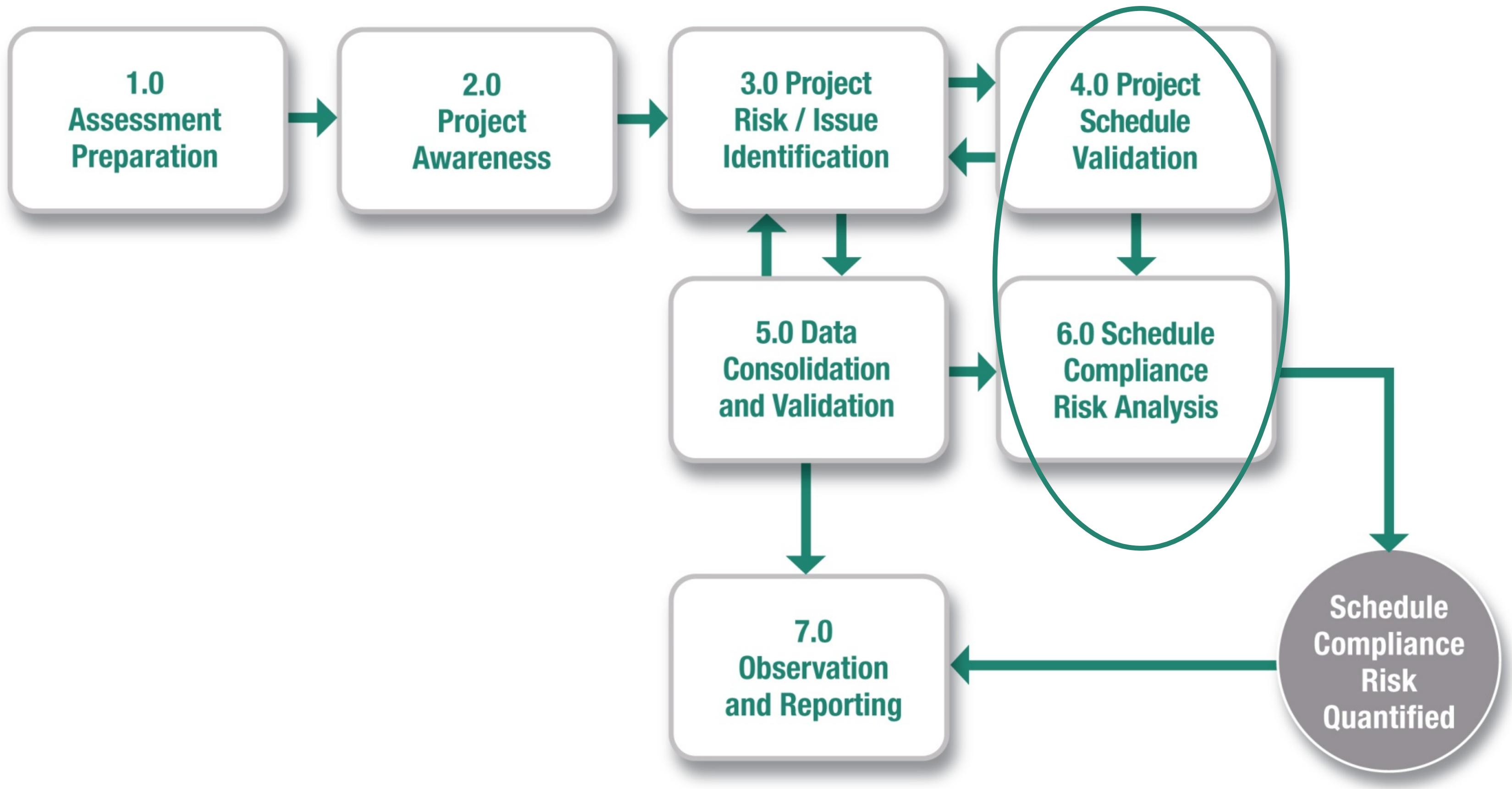
The SCRAM Evaluation Team

- Evaluation conducted by a small team including:
  - Engineers (to validate engineering related BoEs, work load estimates, identify project issues and risks, and provide inputs for schedule risk assessment)
    - Supplemented by domain specific subject matter experts as necessary
    - For software intensive development projects, at least one team member should be proficient in software parametric modelling
  - Scheduler experienced in Project schedule tool
    - Validates schedule – conducts schedule health checks
    - Performs Monte Carlo risk modelling with inputs from engineering team members

SCRAM Assessor Qualification Framework

- Three levels of SCRAM Assessors
  - Provisional SCRAM Assessor
    - Completed SCRAM training and passed exam
  - Certified SCRAM Assessor
    - Participated in SCRAM Evaluations
  - SCRAM Lead Assessor
    - Lead SCRAM Evaluations
- SCRAM Principal
  - Lead SCRAM Evaluations
  - SCRAM Instructors
  - SCRAM Model Developers

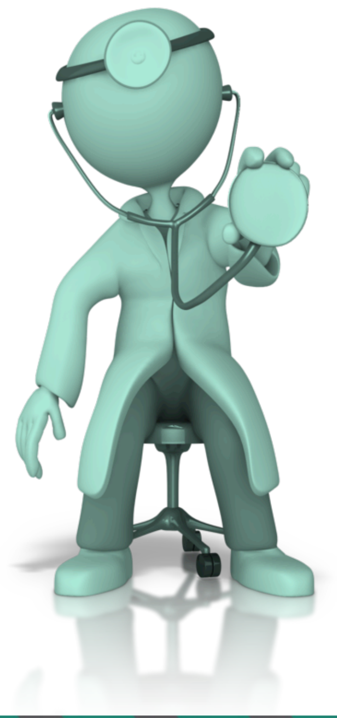
Project Schedule Validation and Analysis


1.0 Assessment Preparation
2.0 Project Awareness
3.0 Project Risk / Issue Identification
4.0 Project Schedule Validation
5.0 Data Consolidation and Validation
6.0 Schedule Compliance Risk Analysis
7.0 Observation and Reporting
Schedule Compliance Risk Quantified

Schedule Health Check

- Performed by the Schedule Specialist

- Some examples include counts of and criteria for:
  - tasks with no predecessor
  - tasks with no successor
  - tasks with no successor or predecessor
  - tasks with a target start date not earlier than
  - tasks with a target finish date not earlier than
  - tasks with a target finish date not later than
  - negative lags
  - negative total float

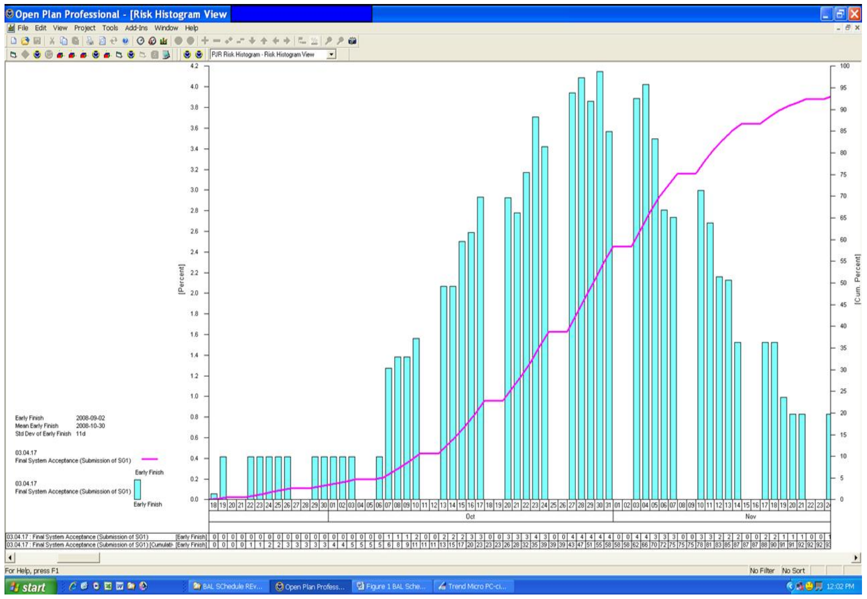


Schedule Risk Analysis/Monte Carlo

- Rate Tasks that are on the Critical or Near Critical Path
  - Assign three point estimates
    - Most Likely, Optimistic and Pessimistic
  - based on identified risks, issues, technical debt and any other sources of delays

- Perform Monte Carlo Simulation
  - provides a picture of the potential impact of risk on schedule

- Projects should use the results of the SRA to develop plans to remediate issues and mitigate risks


Open Plan Professional - [Risk Histogram View]
File Edit View Project Tools Add-Ins Window Help
PJR Risk Histogram - Risk Histogram View
[Percent]
[Cum. Percent]
Oct
Nov
Early Finish
For Help, press F1
start

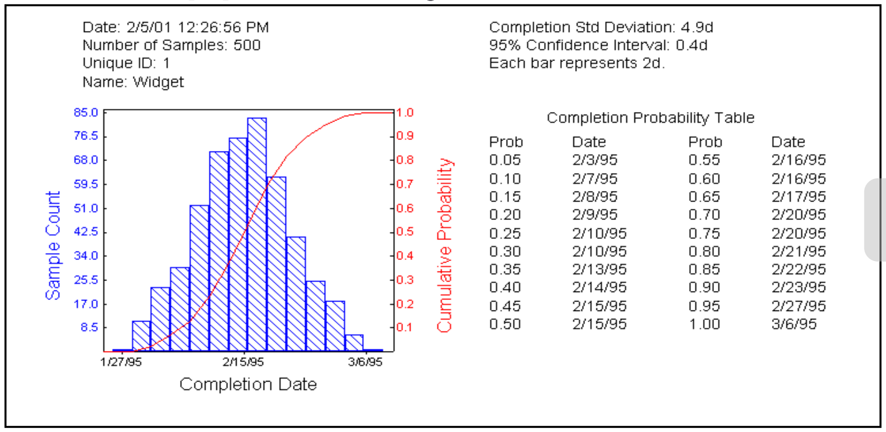
Early Finish 2008-09-02
Mean Early Finish 2008-10-30
Std Dev of Early Finish 11d

03.04.17
Final System Acceptance (Submission of SG1) — Early Finish

03.04.17
Final System Acceptance (Submission of SG1) ▯ Early Finish

SRA Identification of Risk / Opportunity

After risk is introduced, Identifies probabilities of completion if risk is not mitigated



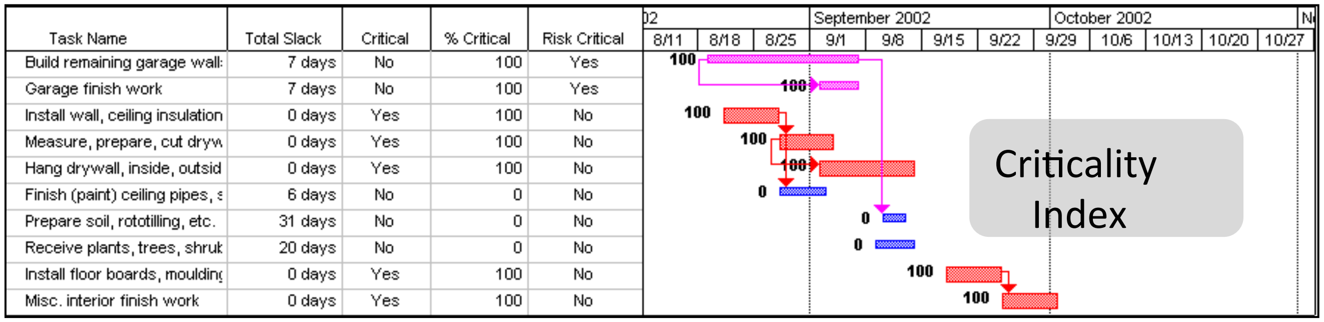
Date: 2/5/01 12:26:56 PM
Number of Samples: 500
Unique ID: 1
Name: Widget

Completion Std Deviation: 4.9d
95% Confidence Interval: 0.4d
Each bar represents 2d.

Completion Probability Table

Prob	Date	Prob	Date
0.05	2/3/95	0.55	2/16/95
0.10	2/7/95	0.60	2/16/95
0.15	2/8/95	0.65	2/17/95
0.20	2/9/95	0.70	2/20/95
0.25	2/10/95	0.75	2/20/95
0.30	2/10/95	0.80	2/21/95
0.35	2/13/95	0.85	2/22/95
0.40	2/14/95	0.90	2/23/95
0.45	2/15/95	0.95	2/27/95
0.50	2/15/95	1.00	3/6/95

Histogram

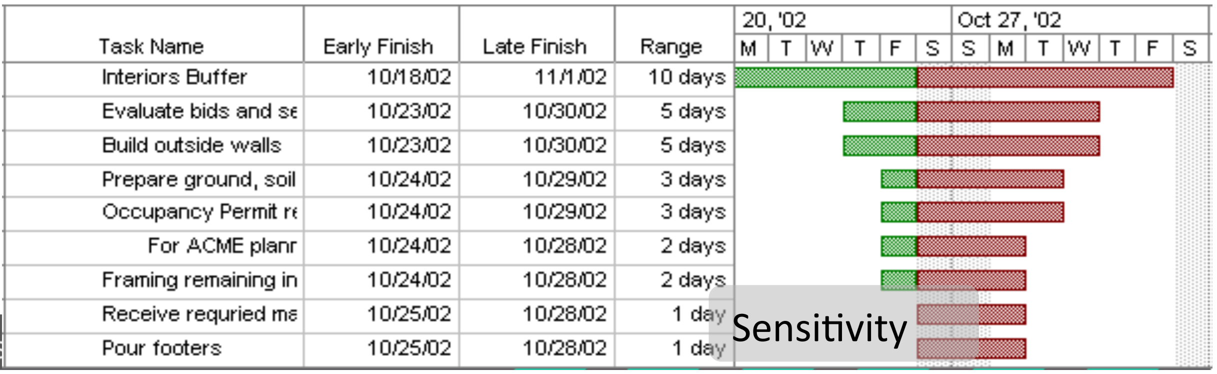


Task Name	Total Slack	Critical	% Critical	Risk Critical
Build remaining garage wall:	7 days	No	100	Yes
Garage finish work	7 days	No	100	Yes
Install wall, ceiling insulation	0 days	Yes	100	No
Measure, prepare, cut dryw	0 days	Yes	100	No
Hang drywall, inside, outsid	0 days	Yes	100	No
Finish (paint) ceiling pipes, ε	6 days	No	0	No
Prepare soil, rototilling, etc.	31 days	No	0	No
Receive plants, trees, shruk	20 days	No	0	No
Install floor boards, moulding	0 days	Yes	100	No
Misc. interior finish work	0 days	Yes	100	No

Criticality Index

After risk is introduced, Identifies tasks that were, and still are critical, newly critical and never critical

After risk is introduced, Identifies tasks most likely to be impacted by mitigation efforts – *"biggest bang for your buck"*

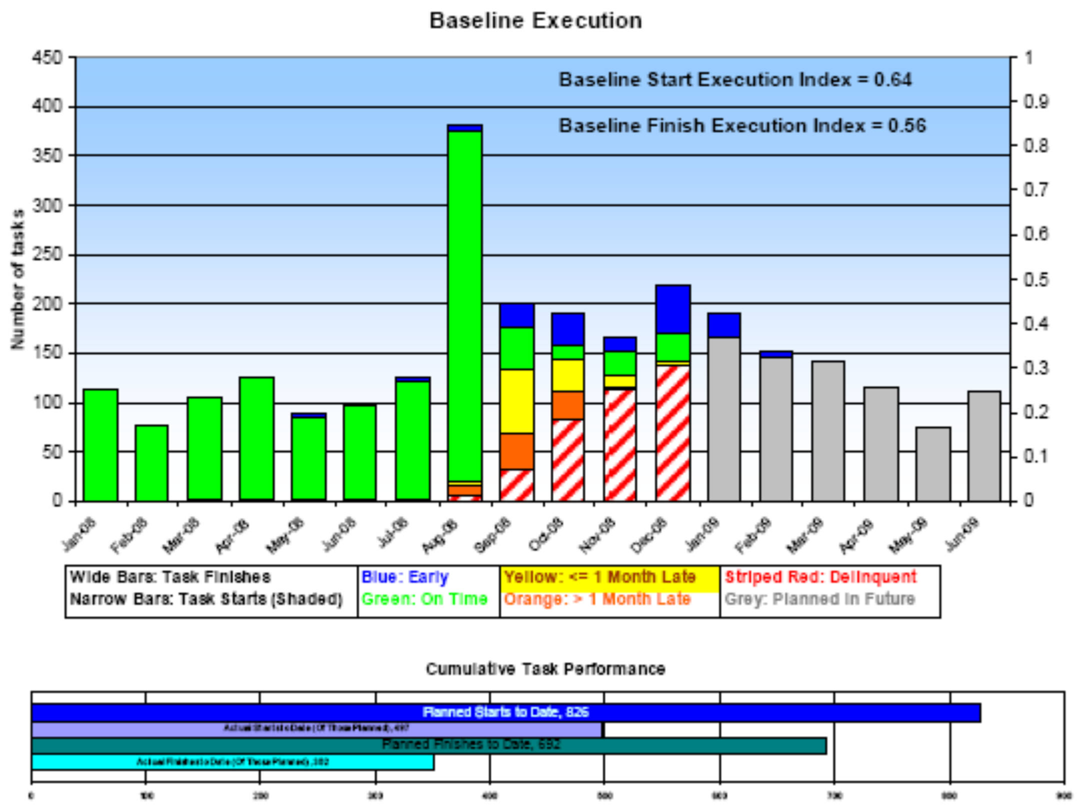


Task Name	Early Finish	Late Finish	Range
Interiors Buffer	10/18/02	11/1/02	10 days
Evaluate bids and se	10/23/02	10/30/02	5 days
Build outside walls	10/23/02	10/30/02	5 days
Prepare ground, soil	10/24/02	10/29/02	3 days
Occupancy Permit re	10/24/02	10/29/02	3 days
For ACME planr	10/24/02	10/28/02	2 days
Framing remaining in	10/24/02	10/28/02	2 days
Receive requried me	10/25/02	10/28/02	1 day
Pour footers	10/25/02	10/28/02	1 day

Sensitivity

Material courtesy of NAVAIR 4.2

Baseline Execution Analysis



Example of a Baseline Execution Chart

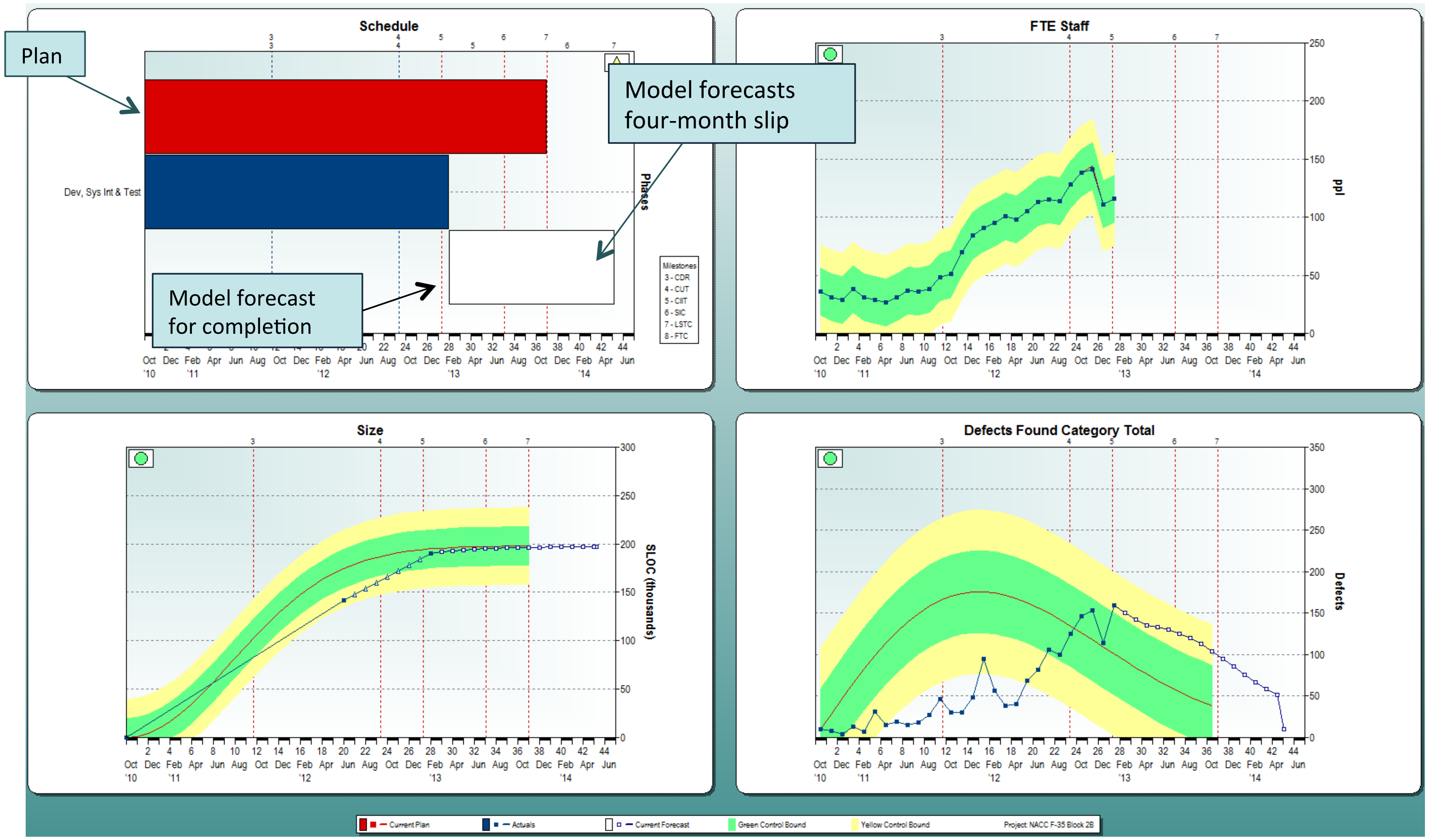
- Presents Contractor's performance to the baseline plan to date
- The larger the number of delinquent tasks, the larger the bow-wave of pending work to complete to regain performance
- Chart represents latest IMS deliverable information (i.e., not built off of previous deliverables)

Material courtesy of NAVAIR 4.2

Parametric Model Forecast

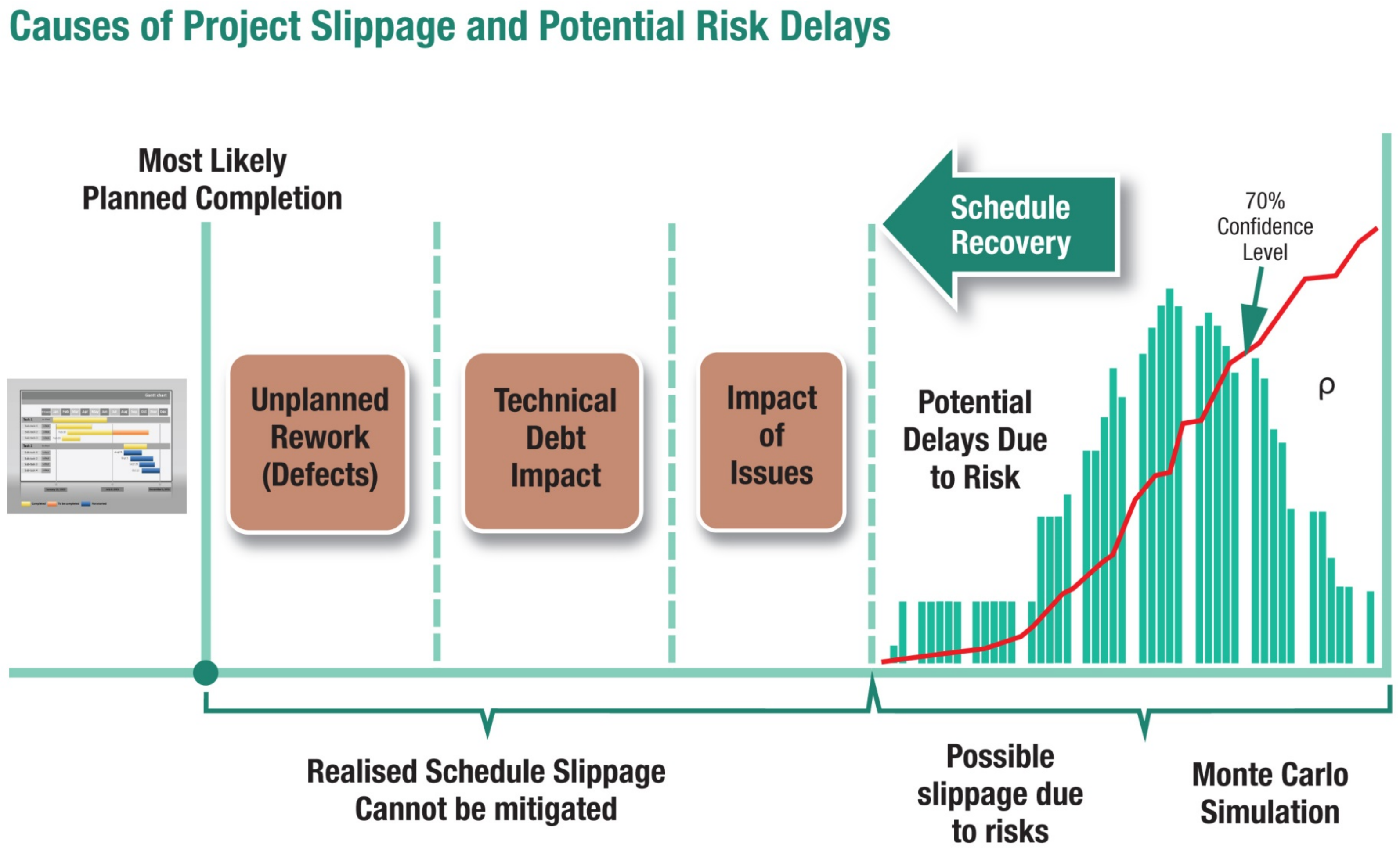
- In addition to conducting a Schedule Risk Analysis, a SCRAM evaluation uses a parametric model to forecast software completion
- A parametric estimation model is a statistical tool with parameters (e.g., estimated size, complexity, programmer experience) to describe the characteristics of a software development.
  - Based on the input parameters, the model estimates duration/ schedule, effort/staffing, and defects.
- SCRAM uses a parametric model that uses actual performance to date to forecast software completion. Inputs include
  - Total size in source lines of code
  - Defects discovered
  - Major milestones completed
  - Staffing

Example Forecast



scram. SCHEDULE COMPLIANCE RISK ASSESSMENT METHODOLOGY

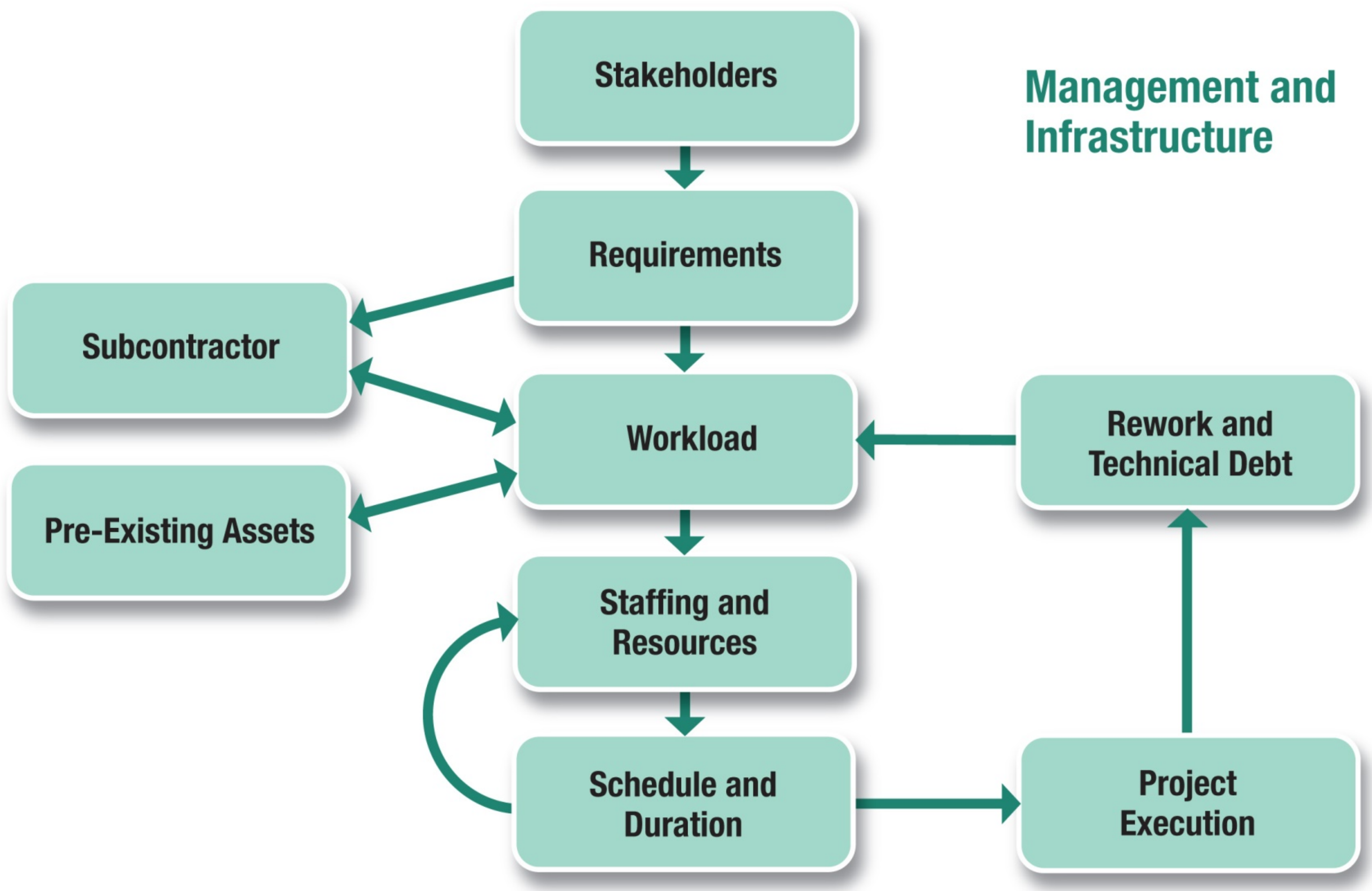
SCRAM Identifies and Quantifies Schedule Slippage Root Causes and Risk


Causes of Project Slippage and Potential Risk Delays
Most Likely Planned Completion
Unplanned Rework (Defects)
Technical Debt Impact
Impact of Issues
Schedule Recovery
70% Confidence Level
ρ
Potential Delays Due to Risk
Realised Schedule Slippage Cannot be mitigated
Possible slippage due to risks
Monte Carlo Simulation

Course Outline

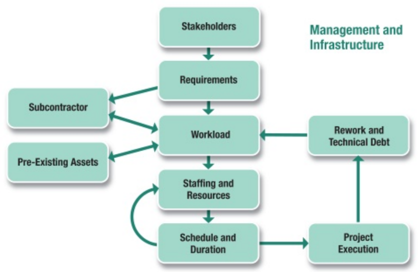
- Participant Introductions
- SCRAM History, Context
- Overview of SCRAM Process
- Overview of the Root Cause Analysis of Schedule Slippage (RCASS) Model
- Overview of the SCRAM Process Reference/Assessment Model
- RCASS Categories and Processes (with exercises)
- Supporting Methods (SRA and Parametric Modelling)
- Wrap Up

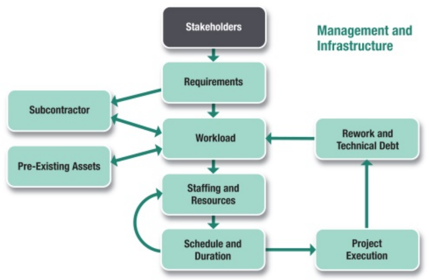
Root Cause Analysis of Schedule Slippage (RCASS) Model



Root Cause Analysis of Schedule Slippage (RCASS) Model

- Has evolved from our experiences in conducting evaluations
- Shows logical dependencies and linkages between information categories
- Covers project planning and project execution
- Used in a SCRAM evaluation as guidance to:
  - Focus and guide questions
  - Categorise the fire hose of data and details gathered during an assessment
  - Ensure complete coverage and highlight missing information
  - Determine the root causes of schedule slippage
  - Identify appropriate measures to serve as leading indicators
    - For visibility and tracking of risks and risk-realisation thresholds





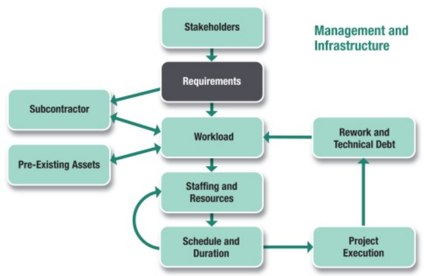
Stakeholders

Reflects project turbulence because of difficulties in synchronising the project's stakeholders: users, customers, system engineers, developers, maintainers, others.

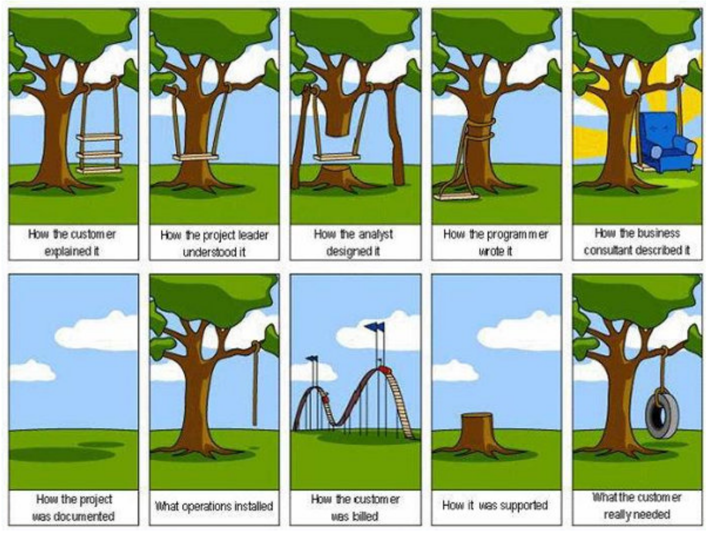
"Our stakeholders are like a 100-headed hydra – everyone can say 'no' and no one can say 'yes'."

SCRAM Examples

- Late in development, a critical stakeholder (customer) added a condition for acceptance that removed three months from the development schedule
- Failed organisational relationship led to breakdown in communication
  - Key stakeholders were not talking to each other even though they were in the same facility
  - Communication degraded to memos slipped under the door



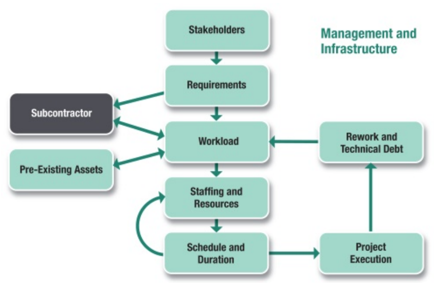
Requirements


How the customer explained it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
How the business consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Reflects understanding and stability of the functional requirements, performance requirements, system constraints, standards, etc. used to define and bound what is to be developed

SCRAM Examples

- Misinterpretation of a communication standard led to an additional 3,000 derived requirements to implement the standard.
- A large ERP project had two system specifications – one with the sponsor/customer and a different specification under contract with the developer – would this be a problem?



Subcontractor

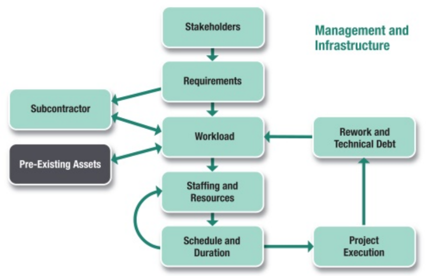
Reflects subcontractor products or services that will be delivered as a part of the overall system.

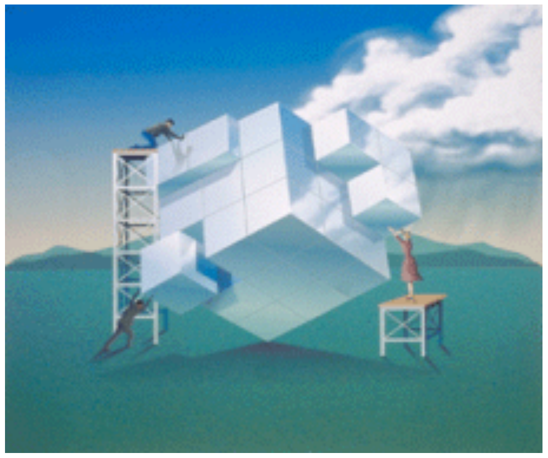
If the subcontractor doesn't perform, additional work required by the Prime

SCRAM Examples

- Subcontractor omitting processes in order to make delivery deadlines led to integration problems with other system components
- Prime and sub-contractor schedule not linked or aligned

Pre-existing Assets





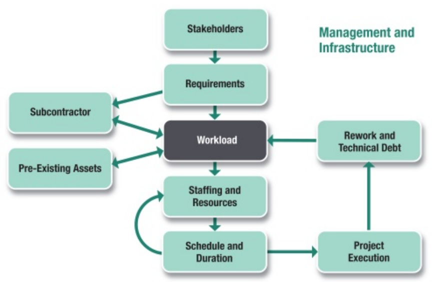
Reflects products developed independent of the project that will be used in the final product, i.e. an asset that reduces the amount of new work that has to be done on a project.

"It doesn't do what we thought…"

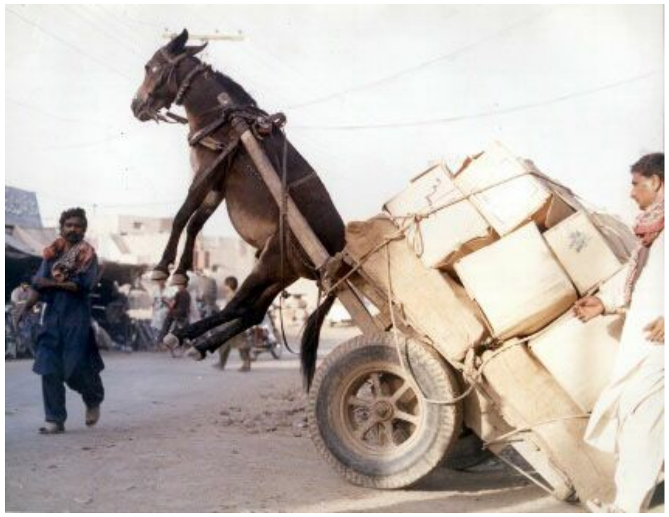
"There is a lot of functionality we don't need."

SCRAM Examples

- COTS products that didn't work as advertised, resulting in additional work
- COTS product required a "technology refresh" during development as the project was years late (cost the project $8M)



Workload



Reflects the quantity of work to be done and provides a basis for estimating effort/staffing and duration

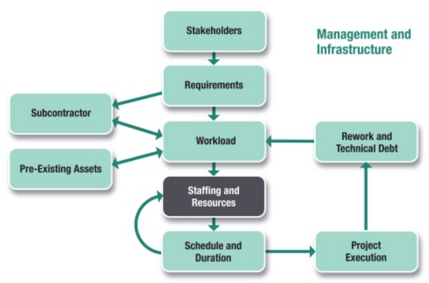
"Unrealistic expectations based on inaccurate estimates are the single largest cause of software failure."

» Futrell, Schafer

SCRAM Examples

- Source lines of code is typically underestimated
- Contract data deliverables workload often underestimated by both contractor and customer
- System Integration labs underestimated resulting in additional cost to build another lab (inadequate basis of estimates)
- Re-plan based on twice the historic productivity with no basis for improvement

Staffing & Resources



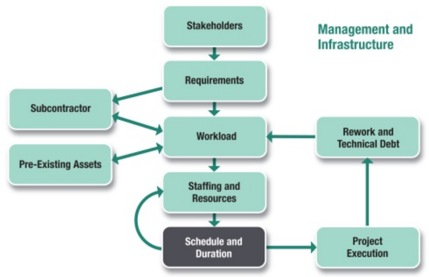
Staff Leaving the Department

Reflects the availability, capability and experience of the staff necessary to do the work as well as the availability and capacity of other resources, such as test and integration labs.

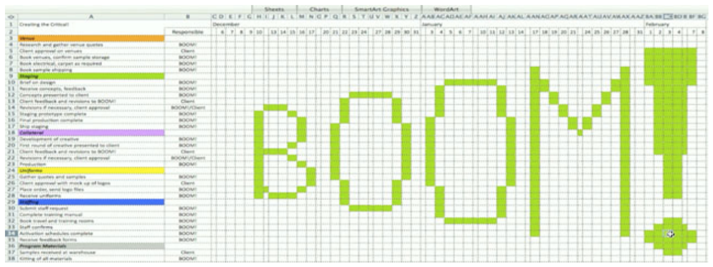
Bringing on people to solve a slippage problem may make it worse (especially late in the project)

SCRAM Examples

- Parent company sacking workers as soon as job is completed
  - Workers went on a “Go Slow”
- Scheduling staff for 12 hours a day (to recover schedule)
- Lack of fidelity and qualification of integration and test lab (resource)



Schedule & Duration

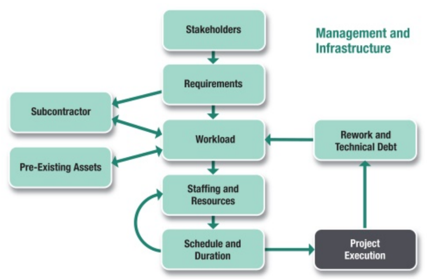


Reflects the task sequencing and calendar time needed to execute the workload by available staff and other resources (e.g. test labs).

SCRAM Examples

- No effective integrated master schedule to provide an overall understanding of the completion date of the project
  - 13 subordinate schedules used to manage project
  - Failed Health Checks
  - Critical Path went subterranean!

Project Execution



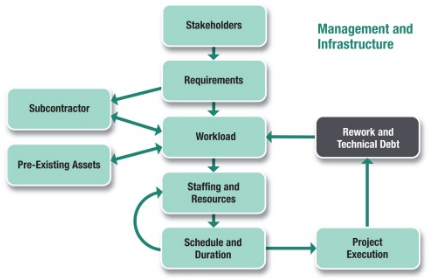
Focuses on monitoring and controlling the execution of the project in accordance with the project schedule

Experience from multiple SCRAM evaluations has highlighted the need to focus on System Integration and Technical Progression

SCRAM Examples

- The schedule was not available to program staff or stakeholders
  - Undergoing a schedule tool transition for approx. 2 years
- Five delivery iterations not scheduled before CDRL approvals
- No System Integration Plan
- No "red" risks on a program undergoing a major contract overrun breach

Rework and Technical Debt



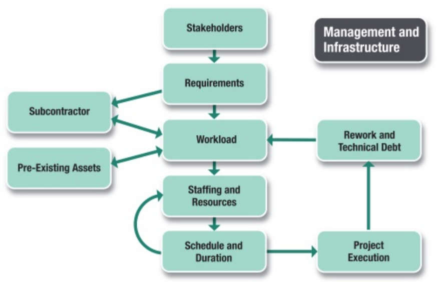
Reflects additional work caused by the discovery of defects in the product and/or associated artefacts, work that is deferred for short-term expediency (Technical Debt) and their resolution

Technical Debt includes suspension of peer reviews, short-cuts in unit test, postponing functionality until later.

Rework is often underestimated or not planned for.

SCRAM Examples

- Suspension of peer reviews led to a bow wave of defects found in System Test
- Accrual of Technical Debt with no repayment plan



Management & Infrastructure

Addresses the factors that impact the efficiency and effectiveness of getting work done, e.g. work processes, use of management and technical software tools, management practices, etc.

Includes processes for Verification and Validation, Infrastructure, Quality Assurance, Process Improvement and Configuration Management

SCRAM Examples

- Change management process and tools could not keep pace with requirements for System Integration and Test
- Lack of software quality assurance on a software intensive project

Course Outline

- Participant Introductions
- SCRAM History, Context
- Overview of SCRAM Process
- Overview of the Root Cause Analysis of Schedule Slippage (RCASS) Model
- Overview of the SCRAM Process Reference/Assessment Model
- RCASS Categories and Processes (with exercises)
- Supporting Methods (SRA and Parametric Modelling)
- Wrap Up

SCRAM Process Reference/Assessment Model

- Provides guidance on the valid construction/preparation of a project schedule for use in project execution
- Developed to support SCRAM Evaluations
- Two models combined to conserve resources
  - SCRAM Process Reference Model
  - SCRAM Process Assessment Model
- ISO/IEC 15504 conformant model

Why conform to ISO/IEC 15504?

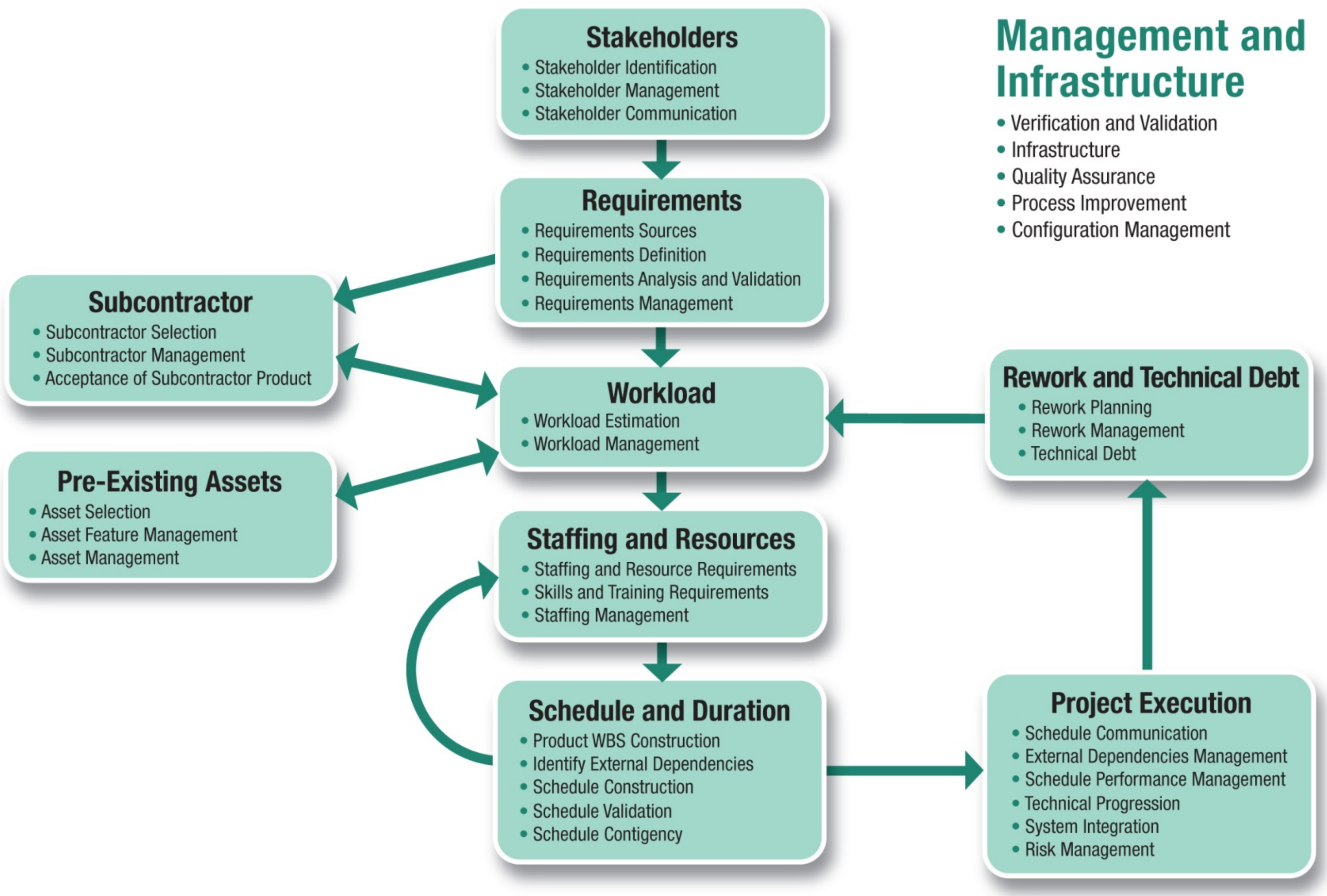
- SCRAM has origins in the conduct of non-advocate reviews of DMO Projects of Interest or Concern
- SCRAM reached a point where repeatable, consistent outcomes were achieved and useful to resolve programmatic and technical issues
- Decision taken to
  - formalise SCRAM for broader use within DMO and Industry
  - align SCRAM with the international standard ISO/IEC15504 Information Technology – Process Assessment framework to increase the validity, credibility and industry acceptance of SCRAM evaluation outcomes
- Aligning SCRAM with ISO/IEC 15504 will not only allow diagnostic evaluation of troubled projects but also a Project's capability [likelihood, potential] of achieving schedule compliance before issues arise

SCRAM Process Reference Model

- 37 processes grouped in 10 Categories
  - structured around the RCASS model

- Every process in the Process Reference Model
  - is uniquely identified
  - has a purpose statement
  - describes a set of outcomes
    - collectively achieve the purpose statement

SCRAM PR/AM Processes



Stakeholders
• Stakeholder Identification
• Stakeholder Management
• Stakeholder Communication
Requirements
• Requirements Sources
• Requirements Definition
• Requirements Analysis and Validation
• Requirements Management
Management and Infrastructure
• Verification and Validation
• Infrastructure
• Quality Assurance
• Process Improvement
• Configuration Management
Subcontractor
• Subcontractor Selection
• Subcontractor Management
• Acceptance of Subcontractor Product
Workload
• Workload Estimation
• Workload Management
Rework and Technical Debt
• Rework Planning
• Rework Management
• Technical Debt
Pre-Existing Assets
• Asset Selection
• Asset Feature Management
• Asset Management
Staffing and Resources
• Staffing and Resource Requirements
• Skills and Training Requirements
• Staffing Management
Schedule and Duration
• Product WBS Construction
• Identify External Dependencies
• Schedule Construction
• Schedule Validation
• Schedule Contigency
Project Execution
• Schedule Communication
• External Dependencies Management
• Schedule Performance Management
• Technical Progression
• System Integration
• Risk Management

Example SCRAM Process

REQ-SRC - Requirements Sources

Purpose

The purpose of Requirement Sources is to ensure that all relevant sources of requirements are considered in eliciting and gathering requirements.

Outcomes

As a result of successful implementation of this process:

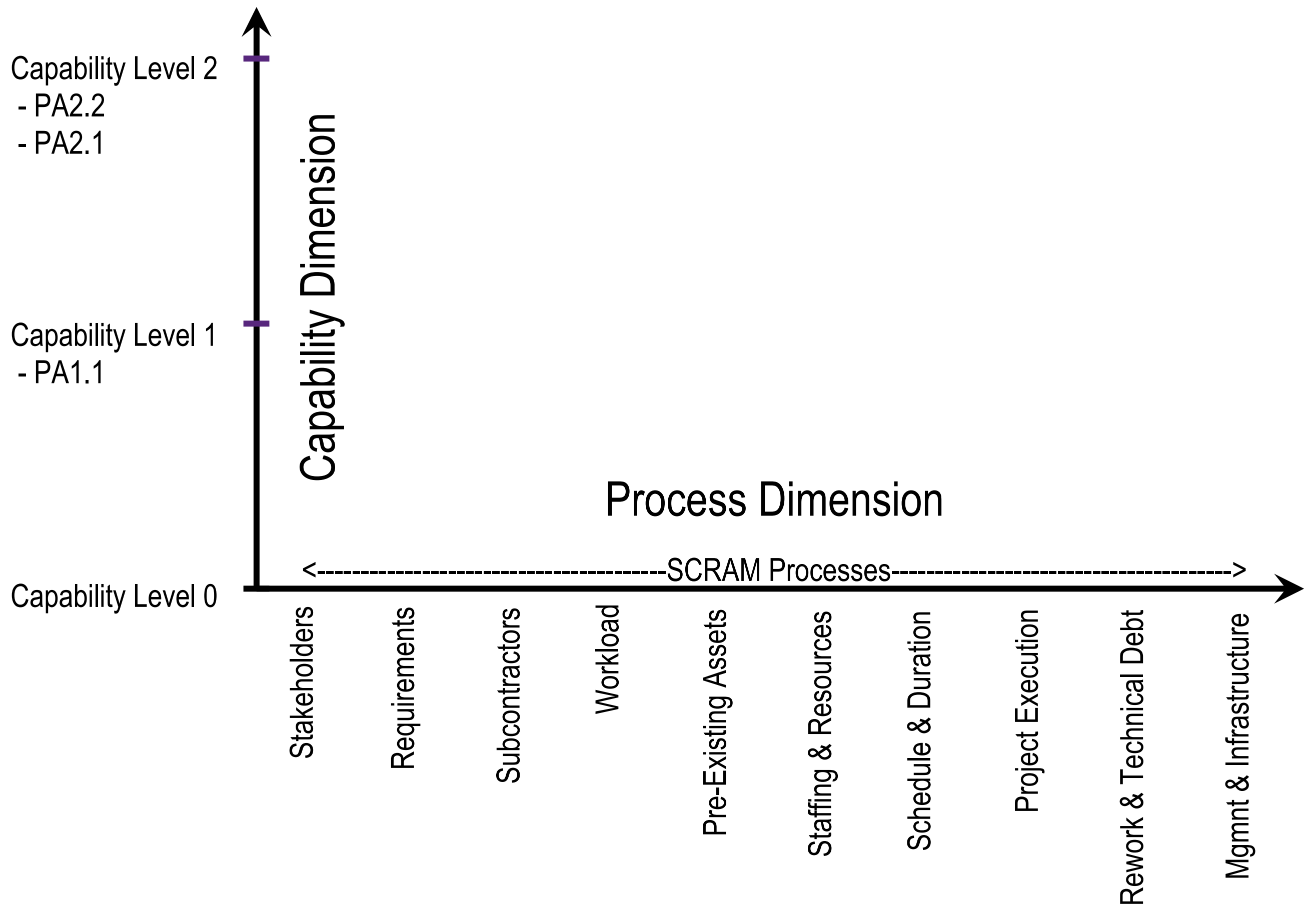
REQ-SRC-1	Capability Definition including Operational Concepts and Environments are established
REQ-SRC-2	Customer/Stakeholder Product Needs, Expectations and Constraints are elicited
REQ-SRC-3	Customer/Stakeholder Requirements are established based on Customer/Stakeholder Product Needs, Expectations and Constraints
REQ-SRC-4	Relevant Legislative and Regulatory Requirements are identified
REQ-SRC-5	System Assurance Requirements are identified
REQ-SRC-6	System Integration and Test Requirements are identified
REQ-SRC-7	Support System Requirements are identified

SCRAM Process Assessment Model

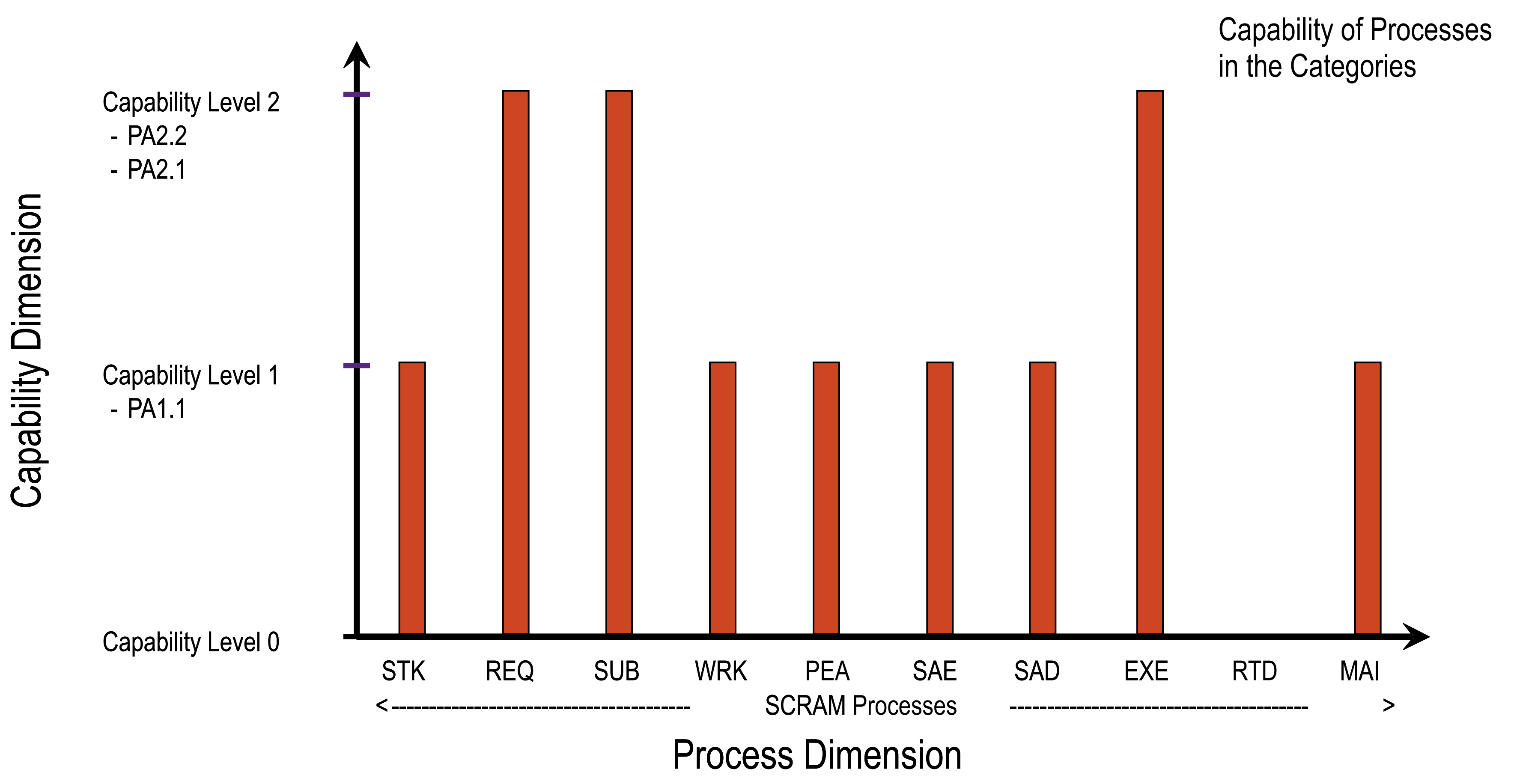
- Describes two (2) levels of capability as defined in ISO/IEC 15504
- Capability Level 1: Performed
  - Base practices
  - Base Work products
- Capability Level 2: Managed
  - Capability practices
  - Capability Work Products

SCRAM Process Assessment Model

- Two-dimensional model of processes and process capability



SCRAM Process Capability Profile



Base Practices & Work Products

- Capability Level 1 indicators that refer to an outcome within the process
- Base Practices
  - provide guidance and information on how to achieve process outcomes
    - used by projects to contribute towards constructing a schedule that minimises the risk of schedule slippage
    - used by SCRAM Teams to identify issues and risks
- Base Work Products
  - a possible artefact resulting from performing the base practice
    - used by projects to record application of the base practice
    - used by SCRAM Teams to seek evidence of the base practice
      - and ultimately identify issues and risks (particularly if the artefact is missing)
- To achieve the process outcome, base practices and base work products are recommended NOT mandatory

Capability Practices & Work Products

- Capability Level 2 indicators that refer to a Process Attribute

- Capability Practices
  - provide guidance and information on achieving higher process capability

- Capability Work Products
  - a possible artefact resulting from performing the capability practice

REQ-SRC Assessment Indicators

Capability Level 1

Process Attribute 1.1 – Process Performance

Example:

Outcome Ref No.		Base Practices	Base Work Products
REQ-SRC-1	BP1	Establish Capability Definition. Capability Definition should include Operational Concepts, Environments and Scenarios, Test Concepts, functionality, performance, maintenance, support and disposal of the systems as appropriate to the scope of the project. Operating environments include systems boundaries and constraints.	Capability Definition Document - Operational Concepts Document - Test Concept document - Function & Performance Specification Model Based System Engineering artefacts Architectural Views - enterprise, operational and technical views

Capability Level 2

Process Attribute 2.1 - Performance management

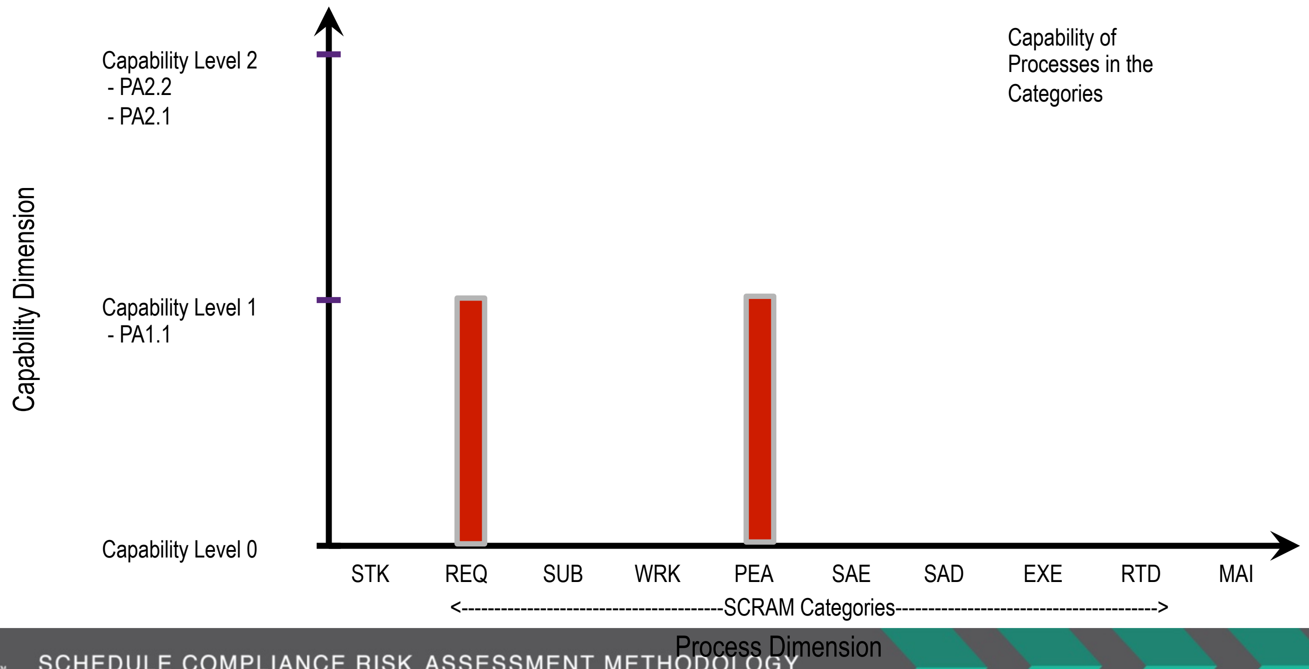
Process Attribute 2.2 - Work product management

Example:

Attribute Achievement		Capability Practices	Capability Work Products
REQ-SRC-2.1a	CP1	Identify the objectives for Requirements Sources. Objectives for performing Requirements Sources include cycle time, resource usage and quality of the work products produced.	Project Plan - with Requirements Sources identified Requirements Plan - with objectives for Requirements Sources - may be part of the Project Plan

Current use of Capability Levels

- To date, SCRAM Evaluations have focussed on Projects of Interest or Concern
  - use a Diagnostic SCRAM (D-SCRAM)
  - the issues and risks driving schedule slippage are generally at Capability Level 1 (or Base Practice Level)
  - and the capability profile would perhaps look something like…..



Alternative View of SCRAM Evaluation Results

- A single bar on a graph is not very helpful
- To get granularity and benefit from current SCRAM Evaluations, the team classifies and reports observations as follows:
  - + Exceptional Strengths
  - ~ Risks
  - ¬ Issues At the Base Practice level
- With this in mind, in this course, we will only focus on Base Practices
- Capability Practices (Level 2)
  - Advanced topic
  - Require validation
  - Future direction of SCRAM

REQ-SRC Base Practices & Work Products

Outcome Ref No.		Base Practices	Base Work Products
REQ-SRC-1	BP1	Establish Capability Definition. Capability Definition should include Operational Concepts, Environments and Scenarios, Test Concepts, functionality, performance, maintenance, support and disposal of the systems as appropriate to the scope of the project. Operating environments include systems boundaries and constraints.	Capability Definition Document - Operational Concepts Document - Test Concept document - Function & Performance Specification Model Based System Engineering artefacts Architectural Views - enterprise, operational and technical views
REQ-SRC-2	BP2	Elicit the customer/stakeholders needs, expectations and constraints for the product. Elicitation actively gathers the customer/stakeholder needs, expectations and constraints for the product being developed and extends beyond simple gathering of requirements. Elicitation methods include demonstrations; questionnaires, interviews, walkthroughs, prototypes, models, brainstorming, market surveys, business cases. Note: This practice focuses on the stakeholder needs and expectations for the product. Refer to Stakeholder (STK) Category Processes for stakeholder needs and expectations for communication and engagement.	Records of Customer/Stakeholder product needs, expectations and constraints
REQ-SRC-3	BP3	Establish the customer/stakeholder requirements based on the customer/stakeholder needs, expectations and constraints for the product. Customer/stakeholder requirements should be traceable to the customer/ stakeholder needs, expectations and constraints for the product.	Customer/Stakeholder Requirements
REQ-SRC-4	BP4	Identify relevant legislative and regulatory requirements. Legislative and regulatory policies often stipulate requirements that may apply during development as well as product requirements. Identification of these policies (and the resulting requirements) ensures the developed product will comply.	List of applicable legislative and regulatory documents including Acts, Policies and Standards Compliance Checklist Product Specifications
REQ-SRC-5	BP5	Identify the requirements for systems assurance. Systems Assurance requirements cover safety, security, software and mission assurance e.g. reliability and dependability	System Assurance Requirements Product Specification
REQ-SRC-6	BP6	Identify System Integration and Test requirements. Define requirements for System Integration and Test Laboratory qualification. Establish a System Integration Plan (SIntP) that defines the strategy and requirements for integration and test processes, plans, facilities, and equipment strings.	System Integration and Test Requirements Document System Integration Plan
REQ-SRC-7	BP7	Identify Support System Requirements. The Support System is the infrastructure and new support elements that enable the System Under Development to be effectively operated and supported so it can meet its operational requirements.	Integrated Logistics Support Plan

SCRAM PR/AM Appendix E

- Contains only
  - Purpose
  - Outcomes
  - Base Practices
  - Base Work Products

- Base Practices and Base Work Products are NOT mandatory but
  - Indicators of achieving
    - SCRAM Process Outcomes and Purpose

- Missing any of these practices could constitute a risk

- Recommendation to print Appendix E for reference